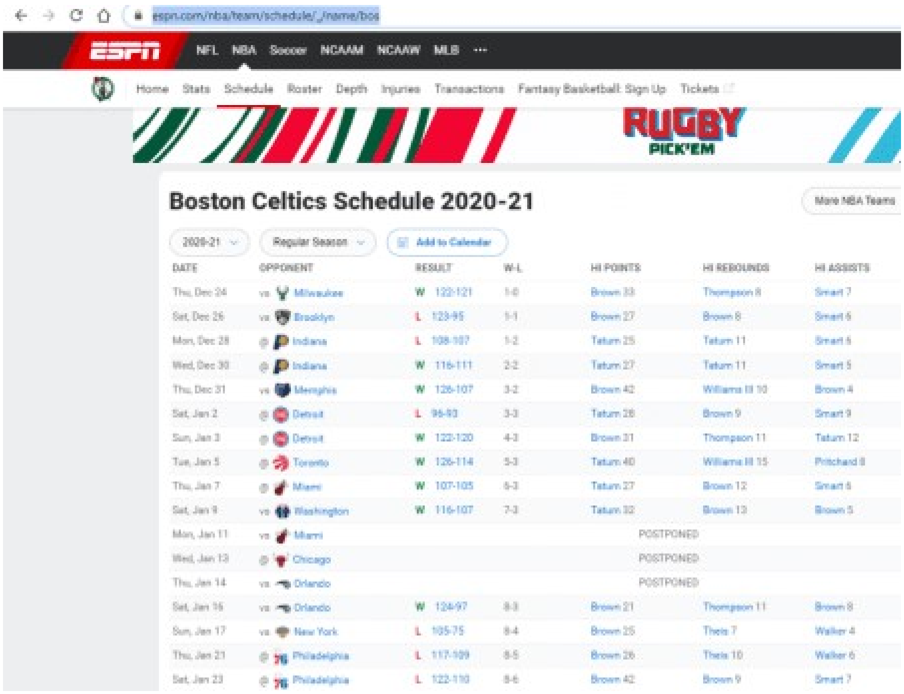
Scrape the Data

We would like to get the results per team. The ESPN URL is of the form https://www.espn.com/nba/team/schedule/_/name/tor where the last part is for the team. So, for the **Toronto Raptors** is `tor` for **Boston Celtics** is `bos` and so on. Let's have a look at the Boston Celtics page:



Actually, we care for the columns `DATE`, `OPPONENT`, `RESULT` and `W-L`. Let's create a script to get the results of all teams and to store them in a data frame called **by_team**. Note that I had to find myself the team codes, such as `tor`, `mil`, `den` and so on.

```
library(rvest)
library(lubridate)
library(tidyverse)
library(stringr)
library(zoo)
library(h2o)
library(lubridate)




teams<-c("tor", "mil", "den", "gs", "ind", "phi", "okc", "por", "bos",
"hou", "lac", "sa",
         "lal", "utah", "mia", "sac", "min", "bkn", "dal", "no", "cha",
"mem", "det", "orl",
         "wsh", "atl", "phx", "ny", "chi", "cle")

teams_fullname<-c("Toronto", "Milwaukee", "Denver", "Golden State",
"Indiana", "Philadelphia", "Oklahoma City","Portland",
                  "Boston", "Houston", "LA", "San Antonio", "Los
Angeles", "Utah", "Miami", "Sacramento", "Minnesota", "Brooklyn",
                  "Dallas", "New Orleans", "Charlotte", "Memphis",
"Detroit", "Orlando", "Washington", "Atlanta", "Phoenix",
```

```
                        "New York", "Chicago", "Cleveland")

by_team<-{}
for (i in 1:length(teams)) {
  url<-paste0("http://www.espn.com/nba/team/schedule/_/name/", teams[i])
  #print(url)
  webpage <- read_html(url)
  team_table <- html_nodes(webpage, 'table')
  team_c <- html_table(team_table, fill=TRUE, header = TRUE)[[1]]
  team_c<-team_c[1:which(team_c$RESULT=="TIME")-1,]
  team_c$URLTeam<-toupper(teams[i])
  team_c$FullURLTeam<-(teams_fullname[i])
  by_team<-rbind(by_team, team_c)
}

# remove the postponed games
by_team<-by_team%>%filter(RESULT!='Postponed')
```

```
> by_team%>%head(10)
           DATE        OPPONENT  RESULT W-L    Hi Points Hi Rebounds  Hi Assists URLTeam FullURLTeam
1  Wed, Dec 23 vsNew Orleans  L113-99 0-1   Siakam  20   Baynes  9    Lowry  10     TOR     Toronto
2  Sat, Dec 26  @San Antonio L119-114 0-2 VanVleet  27   Siakam 15    Lowry  10     TOR     Toronto
3  Tue, Dec 29 @Philadelphia  L100-93 0-3    Lowry  24    Lowry  8    Lowry   9     TOR     Toronto
4  Thu, Dec 31    vsNew York  W100-83 1-3 VanVleet  25  Boucher  9 VanVleet   7     TOR     Toronto
5   Sat, Jan 2  @New Orleans L120-116 1-4 VanVleet  27 VanVleet  8    Lowry   8     TOR     Toronto
6   Mon, Jan 4      vsBoston L126-114 1-5 VanVleet  35 VanVleet  8    Lowry   5     TOR     Toronto
7   Wed, Jan 6      @Phoenix L123-115 1-6   Siakam  32   Siakam  9 VanVleet   7     TOR     Toronto
8   Fri, Jan 8   @Sacramento W144-123 2-6 VanVleet  34 Boucher  10   Siakam  12     TOR     Toronto
9  Sun, Jan 10 @Golden State L106-105 2-7   Siakam  25   Siakam 11    Lowry   6     TOR     Toronto
10 Mon, Jan 11     @Portland L112-111 2-8   Siakam  22   Siakam 13   Siakam  10     TOR     Toronto
```

```
> by_team%>%tail(10)
             DATE         OPPONENT      RESULT   W-L  Hi Points Hi Rebounds  Hi Assists URLTeam FullURLTeam
1057 Fri, Feb 12        @Portland    L129-110 10-17 Sexton  25 Windler  10   Sexton  5     CLE   Cleveland
1058 Sun, Feb 14              @LA    L128-111 10-18 Sexton  22   Allen  10  Garland  6     CLE   Cleveland
1059 Mon, Feb 15    @Golden State     L129-98 10-19 Sexton  23   Allen  14   Sexton  5     CLE   Cleveland
1060 Fri, Feb 19         vsDenver    L120-103 10-20 Sexton  23   Allen  10    Osman  7     CLE   Cleveland
1061 Sun, Feb 21 vsOklahoma City    L117-101 10-21 Sexton  27   Allen  17  Garland  8     CLE   Cleveland
1062 Tue, Feb 23       vsAtlanta    W112-111 11-21 Sexton  29   Allen  14  Garland  8     CLE   Cleveland
1063 Wed, Feb 24       vsHouston     W112-96 12-21  Allen  26   Allen  18 Garland  10     CLE   Cleveland
1064 Sat, Feb 27   @Philadelphia W112-109 OT 13-21 Sexton  28    Wade  12  Garland  9     CLE   Cleveland
1065  Mon, Mar 1         @Houston     W101-90 14-21 Sexton  39   Allen  15   Sexton  8     CLE   Cleveland
1066  Wed, Mar 3       vsIndiana    L114-111 14-22 Sexton  32   Allen  11  Sexton  10     CLE   Cleveland
```

Transform the Data and Feature Engineering

Now, we will need to clean and modify the data so that to able to train the model. This is the most difficult part of Machine Learning Modelling. What we actually need, is the running percentage of wins of each team before the game as well as the final outcome (Win=1, Lost=0). However, we will take into consideration other features such as the percentage of wins in the last 10 games, as well as the percentage of wins when the team plays home and when it plays away. Let's start:

```
by_team_mod<-by_team%>%select(-(`Hi Points`:`Hi Assists`))%>%mutate(
CleanOpponent = str_replace(str_extract(str_replace(OPPONENT,
"^vs",""), "[A-Za-z].+"), " \\*",""),

HomeAway= ifelse(substr(OPPONENT,1,2)=="vs", "Home", "Away"), WL=`W-
L`)%>%
```

```
  separate(WL, c("W", "L"), sep="-")%>%mutate(Tpct=as.numeric(W) /
(as.numeric(L)+as.numeric(W)))%>%mutate(dummy=1,
Outcome=ifelse(substr(RESULT,1,1)=="W",1,0))%>%
  group_by(URLTeam)%>%mutate(Rank = row_number(),
TeamMatchID=paste0(Rank,URLTeam,HomeAway), TLast10=rollapplyr(Outcome,
10, sum, partial = TRUE)/ rollapplyr(dummy, 10, sum, partial =
TRUE))%>%
  group_by(URLTeam, HomeAway)%>%mutate(Rpct=
cumsum(Outcome)/cumsum(dummy), RLast10=rollapplyr(Outcome, 10, sum,
partial = TRUE)/ rollapplyr(dummy, 10, sum, partial = TRUE))%>%
  mutate_at(vars(Rpct, RLast10), funs(lag))%>%group_by(URLTeam)
%>%mutate_at(vars(Tpct, TLast10), funs(lag))%>%na.omit()%>%
  select(TeamMatchID, Rank, DATE, URLTeam, FullURLTeam, CleanOpponent,
HomeAway,Tpct,TLast10 , Rpct, RLast10, Outcome)
```

```
> by_team_mod
# A tibble: 1,006 x 12
# Groups:   URLTeam [30]
   TeamMatchID  Rank DATE        URLTeam FullURLTeam CleanOpponent HomeAway  Tpct TLast10  Rpct RLast10 Outcome
   <chr>       <int> <chr>       <chr>   <chr>       <chr>         <chr>    <dbl>   <dbl> <dbl>   <dbl>   <dbl>
 1 3TORAway        3 Tue, Dec 29 TOR     Toronto     Philadelphia  Away     0       0     0       0           0
 2 4TORHome        4 Thu, Dec 31 TOR     Toronto     New York      Home     0       0     0       0           1
 3 5TORAway        5 Sat, Jan 2  TOR     Toronto     New Orleans   Away     0.25    0.25  0       0           0
 4 6TORHome        6 Mon, Jan 4  TOR     Toronto     Boston        Home     0.2     0.2   0.5     0.5         0
 5 7TORAway        7 Wed, Jan 6  TOR     Toronto     Phoenix       Away     0.167   0.167 0       0           0
 6 8TORAway        8 Fri, Jan 8  TOR     Toronto     Sacramento    Away     0.143   0.143 0       0           1
 7 9TORAway        9 Sun, Jan 10 TOR     Toronto     Golden State  Away     0.25    0.25  0.2     0.2         0
 8 10TORAway      10 Mon, Jan 11 TOR     Toronto     Portland      Away     0.222   0.222 0.167   0.167       0
 9 11TORHome      11 Thu, Jan 14 TOR     Toronto     Charlotte     Home     0.2     0.2   0.333   0.333       1
10 12TORHome      12 Sat, Jan 16 TOR     Toronto     Charlotte     Home     0.273   0.3   0.5     0.5         1
# ... with 996 more rows
```

The `Tpct` and the `TLast10` is the running total win rate up to now and for the last 10 games respectively for the URL team. The `Rpct and the RLast10 is the relevant running total win rate up to now and for the last 10 games respectively for the URL team, whereby` relevant we mean the `home` and the `away`. Please pay attention to the `lag` function that we have used since we want the running total up until the game, without including the outcome of the game, since this is what we try to predict. Otherwise, we would have “data leakage”.

Now, we should convert the `Rpct` and the `RLast10` to `HRpct` and `HRLast10` if they are referred to Home or to `ARpct` and `ARLast10` if they are referred to Away. Let’s do it:

```
df <- data.frame(matrix(ncol = 16, nrow = 0))
x <- c(colnames(by_team_mod), "HRpct", "HRLast10",  "ARpct",
"ARLast10")
colnames(df) <- x


for (i in 1:nrow(by_team_mod)) {
  if(by_team_mod[i,"HomeAway"]=="Home") {
    df[i,c(1:14)]<-data.frame(by_team_mod[i,c(1:12)],
by_team_mod[i,c(10:11)])
  }
  else {

    df[i,c(1:12)]<-by_team_mod[i,c(1:12)]
    df[i,c(15:16)]<-by_team_mod[i,c(10:11)]

  }
}
```

```
# fill the NA values with the previous ones, group by team

df<-df%>%group_by(URLTeam)%>%fill(HRpct , HRLast10, ARpct,  ARLast10,
.direction=c("down"))%>%ungroup()%>%na.omit()%>%filter(Rank>=10)
```

Notice that for the Machine Learning Model, we included the running total of at least 10 games (`filter(Rank>=10)`)

```
> df
# A tibble: 796 x 16
```

	TeamMatchID	Rank	DATE	URLTeam	FullURLTeam	CleanOpponent	HomeAway	Tpct	TLast10	Rpct	RLast10	Outcome	HRpct	HRLast10	ARpct	ARLast10
	<chr>	<int>	<chr>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	10TORAway	10	Mon, Jan 11	TOR	Toronto	Portland	Away	0.222	0.222	0.167	0.167	0	0.5	0.5	0.167	0.167
2	11TORHome	11	Thu, Jan 14	TOR	Toronto	Charlotte	Home	0.2	0.2	0.333	0.333	1	0.333	0.333	0.167	0.167
3	12TORHome	12	Sat, Jan 16	TOR	Toronto	Charlotte	Home	0.273	0.3	0.5	0.5	1	0.5	0.5	0.167	0.167
4	13TORHome	13	Mon, Jan 18	TOR	Toronto	Dallas	Home	0.333	0.4	0.6	0.6	1	0.6	0.6	0.167	0.167
5	14TORHome	14	Wed, Jan 20	TOR	Toronto	Miami	Home	0.385	0.5	0.667	0.667	0	0.667	0.667	0.167	0.167
6	15TORHome	15	Fri, Jan 22	TOR	Toronto	Miami	Home	0.357	0.4	0.571	0.571	1	0.571	0.571	0.167	0.167
7	16TORAway	16	Sun, Jan 24	TOR	Toronto	Indiana	Away	0.4	0.5	0.143	0.143	1	0.571	0.571	0.143	0.143
8	17TORAway	17	Mon, Jan 25	TOR	Toronto	Indiana	Away	0.438	0.6	0.25	0.25	0	0.571	0.571	0.25	0.25
9	18TORHome	18	Wed, Jan 27	TOR	Toronto	Milwaukee	Home	0.412	0.6	0.625	0.625	0	0.625	0.625	0.25	0.25
10	19TORHome	19	Fri, Jan 29	TOR	Toronto	Sacramento	Home	0.389	0.5	0.556	0.556	0	0.556	0.556	0.25	0.25

```
# ... with 786 more rows
```

The final step is to create the **"full_df"** which is an inner join of the "**Home df**" and the "**Away df**".

```
# create the home df
H_df<-df%>%filter(HomeAway=="Home")%>%ungroup()
colnames(H_df)<-paste0("H_", names(H_df))


# create the away df
A_df<-df%>%filter(HomeAway!="Home")%>%ungroup()
colnames(A_df)<-paste0("A_", names(A_df))


Full_df<-H_df%>%inner_join(A_df, by=c("H_CleanOpponent"="A_
FullURLTeam", "H_DATE"="A_DATE"))%>%
  select(H_DATE, H_URLTeam, A_URLTeam, H_Tpct, H_TLast10, H_HRpct,
H_HRLast10, H_ARpct, H_ARLast10,
         A_Tpct, A_TLast10, A_HRpct, A_HRLast10, A_ARpct, A_ARLast10,
H_Outcome)
```

```
> Full_df
# A tibble: 391 x 16
```

	H_DATE	H_URLTeam	A_URLTeam	H_Tpct	H_TLast10	H_HRpct	H_HRLast10	H_ARpct	H_ARLast10	A_Tpct	A_TLast10	A_HRpct	A_HRLast10	A_ARpct	A_ARLast10	H_Outcome
	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Thu, Jan 14	TOR	CHA	0.2	0.2	0.333	0.333	0.167	0.167	0.5	0.6	0.6	0.6	0.5	0.5	1
2	Sat, Jan 16	TOR	CHA	0.273	0.3	0.5	0.5	0.167	0.167	0.462	0.5	0.6	0.6	0.429	0.429	1
3	Mon, Jan 18	TOR	DAL	0.333	0.4	0.6	0.6	0.167	0.167	0.5	0.6	0.667	0.667	0.5	0.5	1
4	Wed, Jan 20	TOR	MIA	0.385	0.5	0.667	0.667	0.167	0.167	0.417	0.4	0.5	0.5	0.2	0.2	0
5	Fri, Jan 22	TOR	MIA	0.357	0.4	0.571	0.571	0.167	0.167	0.462	0.5	0.5	0.5	0.333	0.333	1
6	Wed, Jan 27	TOR	MIL	0.412	0.6	0.625	0.625	0.25	0.25	0.625	0.7	0.75	0.75	0.429	0.429	0
7	Fri, Jan 29	TOR	SAC	0.389	0.5	0.556	0.556	0.25	0.25	0.412	0.4	0.4	0.4	0.333	0.333	0
8	Sun, Jan 31	TOR	ORL	0.368	0.5	0.5	0.5	0.25	0.25	0.4	0.2	0.444	0.444	0.4	0.4	1
9	Sun, Feb 14	TOR	MIN	0.462	0.5	0.545	0.6	0.429	0.6	0.231	0.2	0.364	0.3	0.143	0.1	0
10	Sun, Feb 21	TOR	PHI	0.5	0.7	0.5	0.5	0.471	0.7	0.667	0.6	0.857	0.8	0.467	0.5	1

```
# ... with 381 more rows
```

Build the Predictive Model

Now we are ready to build the Machine Learning model. We will work with the **H2O** library and with the **Random Forest**, although we could have used other algorithms such as **Logistic Regression** etc.

```
# Build the model

h2o.init()
Train_h2o<-as.h2o(Full_df)

Train_h2o$H_Outcome<-as.factor(Train_h2o$H_Outcome)

# random forest model
model1 <- h2o.randomForest(y = 16, x=c(4:15 ), training_frame =
Train_h2o, max_depth=4 )

h2o.performance(model1)
```

```
> h2o.performance(model1)
H2OBinomialMetrics: drf
** Reported on training data. **
** Metrics reported on Out-Of-Bag training samples **

MSE:  0.2474064
RMSE:  0.4973997
LogLoss:  0.6879363
Mean Per-Class Error:  0.4901363
AUC:  0.5536671
AUCPR:  0.6251261
Gini:  0.1073341
R^2:  -0.00290993

Confusion Matrix (vertical: actual; across: predicted) for F1-optimal threshold:
       0   1    Error      Rate
0      5 168 0.971098  =168/173
1      2 216 0.009174    =2/218
Totals 7 384 0.434783  =170/391

Maximum Metrics: Maximum metrics at their respective thresholds
                        metric threshold      value idx
1                       max f1  0.317327   0.717608 383
2                       max f2  0.268171   0.863708 389
3                 max f0point5  0.317327   0.615735 383
4                 max accuracy  0.333494   0.565217 381
5                max precision  0.885988   1.000000   0
6                   max recall  0.268171   1.000000 389
7              max specificity  0.885988   1.000000   0
8             max absolute_mcc  0.671680   0.162528  72
9   max min_per_class_accuracy  0.542183   0.543353 197
10 max mean_per_class_accuracy  0.586147   0.575171 138
11                     max tns  0.885988 173.000000   0
12                     max fns  0.885988 217.000000   0
13                     max fps  0.258071 173.000000 390
14                     max tps  0.268171 218.000000 389
15                     max tnr  0.885988   1.000000   0
16                     max fnr  0.885988   0.995413   0
17                     max fpr  0.258071   1.000000 390
18                     max tpr  0.268171   1.000000 389
```

Make Predictions

The model is ready and we are able to make predictions. We will give as input the Home Team and the Away Team and the algorithm will return the corresponding probabilities of each team to win. What we want is to get the most recent data of each team, which will be the predictors of the model. In order to get the most recent observation by team, we will use the `slice(n())`.

```
#######################
### most recent by team
```

```
########################


### create an empty data frame and fill it in order to get the summary
statistics


df <- data.frame(matrix(ncol = 16, nrow = 0))
x <- c(colnames(by_team_mod), "HRpct", "HRLast10",  "ARpct",
"ARLast10")
colnames(df) <- x


for (i in 1:nrow(by_team_mod)) {
  if(by_team_mod[i,"HomeAway"]=="Home") {
    df[i,c(1:14)]<-data.frame(by_team_mod[i,c(1:12)],
by_team_mod[i,c(10:11)])
  }
  else {

    df[i,c(1:12)]<-by_team_mod[i,c(1:12)]
    df[i,c(15:16)]<-by_team_mod[i,c(10:11)]

  }
}


# fill the NA values with the previous ones group by team

m_df<-df%>%group_by(URLTeam)%>%fill(HRpct , HRLast10, ARpct,  ARLast10,
.direction=c("down"))%>%ungroup()%>%
  na.omit()%>%group_by(URLTeam)%>%slice(n())%>%ungroup()
```

```
> m_df
# A tibble: 30 x 16
   TeamMatchID  Rank DATE        URLTeam FullURLTeam  CleanOpponent HomeAway  Tpct TLast10  Rpct RLast10 Outcome HRpct HRLast10 ARpct ARLast10
   <chr>       <int> <chr>       <chr>   <chr>        <chr>         <chr>    <dbl>   <dbl> <dbl>   <dbl>   <dbl> <dbl>    <dbl> <dbl>    <dbl>
 1 36ATLAway      36 Wed, Mar 3  ATL     Atlanta      Orlando       Away     0.429     0.4 0.389     0.3       1 0.438      0.4 0.389      0.3
 2 37BKNAway      37 Wed, Mar 3  BKN     Brooklyn     Houston       Away     0.639     0.9 0.588     0.7       1 0.722      0.9 0.588      0.7
 3 36BOSHome      36 Thu, Mar 4  BOS     Boston       Toronto       Home     0.514     0.5 0.688     0.7       1 0.688      0.7 0.389      0.3
 4 35CHAAway      35 Wed, Mar 3  CHA     Charlotte    Minnesota     Away     0.471     0.5 0.412     0.4       1 0.5        0.5 0.412      0.4
 5 34CHIAway      34 Wed, Mar 3  CHI     Chicago      New Orleans   Away     0.455     0.6 0.533     0.5       1 0.412      0.5 0.533      0.5
 6 36CLEHome      36 Wed, Mar 3  CLE     Cleveland    Indiana       Home     0.4       0.4 0.5       0.4       0 0.5        0.4 0.25       0.1
 7 34DALHome      34 Wed, Mar 3  DAL     Dallas       Oklahoma City Home     0.515     0.8 0.5       0.6       1 0.5        0.6 0.5        0.5
 8 36DENAway      36 Thu, Mar 4  DEN     Denver       Indiana       Away     0.571     0.6 0.579     0.4       1 0.6        0.7 0.579      0.4
 9 36DETAway      36 Thu, Mar 4  DET     Detroit      New York      Away     0.286     0.4 0.211     0.3       0 0.4        0.4 0.211      0.3
10 37GSAway       37 Thu, Mar 4  GS      Golden State Phoenix       Away     0.528     0.5 0.389     0.4       0 0.647      0.7 0.389      0.4
# ... with 20 more rows
```

Let’s get the predictions of the following 5 games:

Thursday, March 11

MATCHUP			TIME (EET)	NAT TV	TICKETS
Detroit	at	Charlotte	2:00 AM		
Boston	at	Brooklyn	2:30 AM	TNT	Tickets as low as $160
Atlanta	at	Toronto	2:30 AM		
Orlando	at	Miami	3:00 AM		Tickets as low as $22
Philadelphia	at	Chicago	3:00 AM		

```
### Make predictions

df<-{}
a<-c("DET", "BOS", "ATL", "ORL", "PHI")
h<-c("CHA","BKN", "TOR", "MIA", "CHI")

for (i in 1:length(a)) {



  th<-m_df%>%filter(URLTeam==h[i])%>%select(Tpct:ARLast10, -Outcome)
  colnames(th)<-paste0("H_", colnames(th))

  ta<-m_df%>%filter(URLTeam==a[i])%>%select(Tpct:ARLast10, -Outcome)
  colnames(ta)<-paste0("A_", colnames(ta))


  pred_data<-cbind(th,ta)



  tmp<-data.frame(Away=a[i], Home=h[i],as.data.frame(
predict(model1,as.h2o(pred_data))))
  df<-rbind(df, tmp)

}

df<-df%>%select(-predict)
df
```

```
> df
  Away Home        p0        p1
1  DET  CHA 0.2950893 0.7049107
2  BOS  BKN 0.4092680 0.5907320
3  ATL  TOR 0.4117390 0.5882610
4  ORL  MIA 0.4302941 0.5697059
5  PHI  CHI 0.5435127 0.4564873
```

So, according to the model, the DET has 29.5% chances to win against CHA and BOS 40.9% to win against BKN and so on.

Final Thoughts

This is a relatively simple model. We can enrich it by taking into account other features such as the injuries, the days between two games, the traveling distance of the teams and so.